UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/807,075	03/23/2004	Michael J. Azevedo	IBMS.072PA(0513)	2651

62627 7590 06/28/2007

DAVID W. LYNCH
CHAMBLISS, BAHNER & STOPHEL
1000 TALLAN SQUARE-S
TWO UNION SQUARE
CHATTANOOGA, TN 37402

EXAMINER	
PUENTE, EMERSON C	
ART UNIT	PAPER NUMBER
2113	

MAIL DATE	DELIVERY MODE
06/28/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/807,075	AZEVEDO ET AL.	
	Examiner	Art Unit	
	Emerson C. Puente	2113	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *05 April 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *29-56* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *29-56* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *3/23/04* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

This action is made **Final**.

Claims 1-28 have been cancelled. Claims 29-56 have been examined. Examiner notes applicant's amendment necessitated new ground(s) of rejection presented in this Office Action.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 29-56 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 29 recites the limitation "the local processor" in lines 8-9 of claim. There is insufficient antecedent basis for this limitation in the claim. Examiner is uncertain whether the limitation is in reference to the "processor interface". If so, examiner suggests amending to "the processor interface". If not, examiner suggests amending to "a local processor".

Claim 35 recites the limitation "the local processor" in line 7-8 of claim. There is insufficient antecedent basis for this limitation in the claim. Examiner is uncertain whether the limitation is in reference to the "processor interface". If so, examiner suggests amending to "the processor interface". If not, examiner suggests amending to "a local processor".

Claim 41 recites the limitation "the local processor" in line 9 of claim. There is insufficient antecedent basis for this limitation in the claim. Examiner is uncertain whether the

limitation is in reference to the "processor interface". If so, examiner suggests amending to "the processor interface". If not, examiner suggests amending to "a local processor".

Claim 47 recites the limitation "the local processor" in line 7 of claim. There is insufficient antecedent basis for this limitation in the claim. Examiner is uncertain whether the limitation is in reference to the "processor interface". If so, examiner suggests amending to "the processor interface". If not, examiner suggests amending to "a local processor".

Claim 53 recites the limitation "the local processor" in line 6-7 of claim. There is insufficient antecedent basis for this limitation in the claim. Examiner is uncertain whether the limitation is in reference to the "processor interface". If so, examiner suggests amending to "the processor interface". If not, examiner suggests amending to "a local processor".

Claim 54 recites the limitation "the local processor" in line 5-6 of claim. There is insufficient antecedent basis for this limitation in the claim. Examiner is uncertain whether the limitation is in reference to the "processor interface". If so, examiner suggests amending to "the processor interface". If not, examiner suggests amending to "a local processor".

Claim 55 recites the limitation "the local processor" in line 8 of claim. There is insufficient antecedent basis for this limitation in the claim. Examiner is uncertain whether the limitation is in reference to the "processor interface". If so, examiner suggests amending to "the processor interface". If not, examiner suggests amending to "a local processor".

Claim 56 recites the limitation "the local processor" in line 7 of claim. There is insufficient antecedent basis for this limitation in the claim. Examiner is uncertain whether the limitation is in reference to the "processor interface". If so, examiner suggests amending to "the processor interface". If not, examiner suggests amending to "a local processor".

The remaining claims, not specifically mentioned, are rejected because they are dependent upon one of the claims above.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 29-31,34-37,40-43,46-49, and 52-56 are rejected under 35 U.S.C. 102(b) as being anticipated by US Patent No. 5,345,392 of Mito et al. referred hereinafter Mito.

In regards to claims 29 and 35, Mito discloses a program storage device readable by a computer, the program storage device tangibly embodying one or more programs of instructions executable by the computer to perform operations for determining when to perform an error recovery instruction, the operations comprising:

receiving an error recovery instruction. Mito discloses receiving an interrupt (see figure 4 and column 8 lines 36-37).

beginning a timeout task. Mito discloses a watchdog timer (see figure 4 item 172 and column 8 lines 36-37).

monitoring a processor interface to identify processor status for determining a time to perform the error recovery instruction for withholding access to the local processor. Mito discloses monitoring for interrupts from a power management processor interface, indicating a

processor interface, to initiate a suspend routine, indicating withholding access to a local processor (see figure 2,4 and column 8 lines 25-30 and 45-47).

performing the error recovery instruction when the monitoring determines a time for performing the error recovery instruction. Mito discloses initiating a suspend routine (see column 8 lines 45-47).

In regards to claim 30 and 36, Mito discloses the claim limitations as discussed above. Mito further discloses forcing an execution of the error recovery instruction when the timeout task expires before the monitoring determines a time to perform the error recovery instruction. Mito discloses high temperatures and low battery, which could represent time to perform error recovery instructions (see figure 4 items 174,176 and column 8 lines 25-30). Mito further discloses a watchdog timeout, indicating a timeout task (see figure 4 item 172 and column 8 lines 25-30). When the suspend is the result of the watchdog timeout, the handler forces an execution of the error recovery instruction when the timeout task expires before the monitoring determines a time to perform the error recovery instruction.

In regards to claim 31,34,37, and 40, Mito discloses the claim limitations as discussed above. Mito further discloses resuming normal operations after performing the error recovery instruction. Mito discloses a resume handler (see figure 7 and column 9 lines 15-27)

In regards to claim 41, Mito discloses an apparatus for quiescing processor control logic upon receipt of an error recovery instruction, comprising:

self-quiesce logic for receiving an error recovery instruction. Mito discloses receiving an interrupt (see figure 4 and column 8 lines 36-37).

a timer, coupled to the self-quiesce logic, for determining when to force execution of the error recovery instruction. Mito discloses a watchdog timeout (see figure 4 item 172 and column 8 lines 36-37).

wherein the self-quiesce logic initiates the timer when the error recovery instruction is received, begins to monitor a processor interface to identify processor status for determining a time to perform the error recovery instruction for withholding access to the local processor. Mito discloses monitoring for interrupts from a power management processor interface, indicating a processor interface, to initiate a suspend routine, indicating withholding access to a local processor (see figure 2,4 and column 8 lines 25-30 and 45-47).

performs the error recovery instruction when the monitoring determines a time for performing the error recovery instruction. Mito discloses initiating a suspend routine (see column 8 lines 45-47).

In regards to claim 42, Mito discloses the claim limitations as discussed above. Mito further discloses the self-quiesce logic forces an execution of the error recovery instruction when the timeout task expires before the monitoring determines a time to perform the error recovery instruction. Mito discloses high temperatures and low battery, which could represent time to perform error recovery instructions (see figure 4 items 174,176 and column 8 lines 25-30). Mito further discloses a watchdog timeout, indicating a timeout task (see figure 4 item 172 and column 8 lines 25-30). When the suspend is the result of the watchdog timeout, the handler forces an execution of the error recovery instruction when the timeout task expires before the monitoring determines a time to perform the error recovery instruction.

In regards to claims 43 and 46, Mito discloses the claim limitations as discussed above. Mito further discloses wherein the self-quiesce logic allows resuming normal operations after performing the error recovery instruction. Mito discloses a resume handler (see figure 7 and column 9 lines 15-27).

In regards to claim 47, Mito discloses an apparatus for quiescing processor control logic upon receipt of an error recovery instruction, comprising:

a processor for executing instructions (see column 5 lines 26-30).

self-quiesce logic, coupled to the processor, the self-quiesce logic detecting an error recovery instruction, wherein the self-quiesce logic monitors a processor interface to identify processor status for determining a time to perform the error recovery instruction for withholding access to the local processor. Mito discloses monitoring for interrupts from a power management processor interface, indicating a processor interface, to initiate a suspend routine, indicating withholding access to a local processor (see figure 2,4 and column 8 lines 25-30 and 45-47).

performs the error recovery instruction when the monitoring determines a time for performing the error recovery instruction. Mito discloses initiating a suspend routine (see column 8 lines 45-47).

In regards to claim 48, Mito discloses the claim limitations as discussed above. Mito further discloses the self-quiesce logic forces an execution of the error recovery instruction when the timeout task expires before the monitoring determines a time to perform the error recovery instruction. Mito discloses high temperatures and low battery, which could represent time to perform error recovery instructions (see figure 4 items 174,176 and column 8 lines 25-30). Mito

further discloses a watchdog timeout, indicating a timeout task (see figure 4 item 172 and column 8 lines 25-30). When the suspend is the result of the watchdog timeout, the handler forces an execution of the error recovery instruction when the timeout task expires before the monitoring determines a time to perform the error recovery instruction.

In regards to claims 49 and 52, Mito discloses the claim limitations as discussed above. Mito further discloses wherein the self-quiesce logic allows resuming normal operations after performing the error recovery instruction. Mito discloses a resume handler (see figure 7 and column 9 lines 15-27).

In regards to claims 53 and 54, Mito discloses a method for determining when to perform an error recovery instruction, comprising:

receiving an error recovery instruction. Mito discloses receiving an interrupt (see figure 4 and column 8 lines 36-37)

beginning a timeout task. Mito discloses a watchdog timeout (see figure 4 item 172 and column 8 lines 36-37).

monitoring a processor interface to identify processor status for determining a time to perform the error recovery instruction for withholding access to the local processor. Mito discloses monitoring for interrupts from a power management processor interface, indicating a processor interface, to initiate a suspend routine, indicating withholding access to a local processor (see figure 2,4 and column 8 lines 25-30 and 45-47).

performing the error recovery instruction when the monitoring determines a time for performing the error recovery instruction. Mito discloses initiating a suspend routine (see column 8 lines 45-47).

In regards to claim 55, Mito discloses an apparatus for quiescing processor control logic upon receipt of an error recovery instruction, comprising:

means for receiving an error recovery instruction. Mito discloses receiving an interrupt (see figure 4 and column 8 lines 36-37)

means for determining when to force execution of the error recovery instruction. Mito discloses initiating a suspend routine (see column 8 lines 45-47).

wherein the means for receiving an error recovery instruction initiates a timer when the error recovery instruction is received. Mito discloses a watchdog timer (see figure 4 item 172 and column 8 lines 36-37).

begins to monitor a processor interface to identify processor status for determining a time to perform the error recovery instruction for withholding access to the local processor. Mito discloses monitoring for interrupts from a power management processor interface, indicating a processor interface, to initiate a suspend routine, indicating withholding access to a local processor (see figure 2,4 and column 8 lines 25-30 and 45-47).

performs the error recovery instruction when a time for performing the error recovery instruction is determined. Mito discloses initiating a suspend routine (see column 8 lines 45-47).

In regards to claim 56, Mito discloses an apparatus for quiescing processor control logic upon receipt of an error recovery instruction, comprising:

means for executing instructions (see column 5 lines 26-30).

means, coupled to the means for executing instructions, for detecting an error recovery instruction. Mito discloses detecting an interrupt (see figure 4 and column 8 lines 36-37).

monitoring a processor interface to identify processor status for determining a time to perform the error recovery instruction for withholding access to the local processor. Mito discloses monitoring for interrupts from a power management processor interface, indicating a processor interface, to initiate a suspend routine, indicating withholding access to a local processor (see figure 2,4 and column 8 lines 25-30 and 45-47).

performing the error recovery instruction when a time for performing the error recovery instruction is determined. Mito discloses initiating a suspend routine (see column 8 lines 45-47).

Claim 35,38-40,47,50-52,54, and 56 rejected under 35 U.S.C. 102(b) as being anticipated by US Patent No. 4,974,147 of Hanrahan et al. referred hereinafter Hanrahan.

In regards to claim 35, Hanrahan discloses a program storage device readable by a computer, the program storage device tangibly embodying one or more programs of instructions executable by the computer to perform operations for determining when to perform an error recovery instruction, the operations comprising:

receiving an error recovery instruction. Hanrahan discloses a receiving a quiesce signal (see column 9 line 13-14).

monitoring a processor interface to identify processor status for determining a time to perform the error recovery instruction for withholding access to the local processor. Hanrahan discloses determining when there are no longer bus requests (see column 7 lines 20-23), indicating monitoring a processor interface, and performing no further activity (see column 7 lines 20-23), indicating withholding access to the local processor.

performing the error recovery instruction when the monitoring determines a time for performing the error recovery instruction. Hanrahan discloses sequencing to a known state for retry, recover, and debug operations (see column 2 lines 24-26).

In regards to claim 38, Hanrahan discloses the claim limitations as discussed above. Hanrahan further discloses

monitoring a processor interface to a host bus for an idle condition. Hanrahan discloses determining when there are no longer bus requests (see column 7 lines 20-23).

withholding access to the processor interface when the idle condition is detected. Hanrahan discloses performing no further bus activity (see column 7 lines 20-23).

after access to the processor interface is withheld, interrogating all data transfer paths to determine when all the data paths are idle. Hanrahan discloses waiting for all operations to complete before sequencing to a known state (see column 9 lines 15-20).

identifying the time to perform the error recovery instruction when all data transfer paths are idle. Hanrahan discloses sequencing to a known state when all current operations that have completed (see column 9 lines 15-22).

In regards to claim 39, Hanrahan discloses the claim limitations as discussed above. Hanrahan further discloses resuming normal operations after performing the error recovery

instruction. Hanrahan discloses resuming operations following the quiesce state (see column 9 lines 24-26).

In regards to claim 40, Hanrahan discloses the claim limitations as discussed above. Hanrahan further discloses resuming normal operations after performing the error recovery instruction. Hanrahan discloses resuming operations following the quiesce state (see column 9 lines 24-26).

In regards to claim 47, Hanrahan discloses an apparatus for quiescing processor control logic upon receipt of an error recovery instruction, comprising:

a processor for executing instructions (see column 3 lines 17-20).

self-quiesce logic, coupled to the processor, the self-quiesce logic detecting an error recovery instruction. Hanrahan discloses a receiving a quiesce signal (see column 9 lines 13-14).

wherein the self-quiesce logic monitors a processor interface to identify processor status for determining a time to perform the error recovery instruction for withholding access to the local processor. Hanrahan discloses determining when there are no longer bus requests (see column 7 lines 20-23), indicating monitoring a processor interface, and performing no further activity (see column 7 lines 20-23), indicating withholding access to the local processor.

performs the error recovery instruction when the monitoring determines a time for performing the error recovery instruction. Hanrahan discloses sequencing to a known state for retry, recover, and debug operations (see column 2 lines 24-26).

In regards to claim 50, Hanrahan discloses the claim limitations as discussed above. Hanrahan further discloses

wherein the self-quiesce logic monitors a processor interface to a host bus to identify processor status for determining a time to perform the error recovery instruction for withholding access to the local processor by monitoring a processor interface for an idle condition. Hanrahan discloses determining when there are no longer bus requests (see column 7 lines 20-23).

withholding access to the processor interface when the idle condition is detected. Hanrahan discloses performing no further activity (see column 7 lines 20-23).

after access to the processor interface is withheld, interrogating all data transfer paths to determine when all the data paths are idle. Hanrahan discloses waiting for all operations to complete before sequencing to a known state (see column 9 lines 15-20).

identifying the time to perform the error recovery instruction when all data transfer paths are idle. Hanrahan discloses sequencing to a known state when all current operations that have completed (see column 9 lines 15-22).

In regards to claim 51, Hanrahan discloses the claim limitations as discussed above. Hanrahan further discloses wherein the self-quiesce logic causes normal operations to be resumed after performing the error recovery instruction. Hanrahan discloses resuming operations following the quiesce state (see column 9 lines 24-26).

In regards to claim 52, Hanrahan discloses the claim limitations as discussed above. Hanrahan further discloses wherein the self-quiesce logic causes normal operations to be resumed after performing the error recovery instruction. Hanrahan discloses resuming operations following the quiesce state (see column 9 lines 24-26).

In regards to claim 54, Hanrahan discloses a method for determining when to perform an error recovery instruction, comprising:

receiving an error recovery instruction. Hanrahan discloses a receiving a quiesce signal (see column 9 lines 13-14).

monitoring processor interface to identify processor status for determining a time to perform the error recovery instruction for withholding access to the local processor. Hanrahan discloses determining when there are no longer bus requests (see column 7 lines 20-23), indicating monitoring a processor interface, and performing no further activity (see column 7 lines 20-23), indicating withholding access to the local processor.

performing the error recovery instruction when the monitoring determines a time for performing the error recovery instruction. Hanrahan discloses sequencing to a known state for retry, recover, and debug operations (see column 2 lines 24-26).

In regards to claim 56, Hanrahan discloses an apparatus for quiescing processor control logic upon receipt of an error recovery instruction, comprising:

means for executing instructions (see column 3 lines 17-20).

means, coupled to the means for executing instructions, for detecting an error recovery instruction Hanrahan discloses a receiving a quiesce signal (see column 9 lines 13-14).

monitoring a processor interface to identify processor status for determining a time to perform the error recovery instruction for withholding access to the local processor. Hanrahan discloses determining when there are no longer bus requests (see column 7 lines 20-23),

indicating monitoring a processor interface, and performing no further activity (see column 7 lines 20-23), indicating withholding access to the local processor.

performing the error recovery instruction when a time for performing the error recovery instruction is determined. Hanrahan discloses sequencing to a known state for retry, recover, and debug operations (see column 2 lines 24-26).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 29-34,36,37,41-46,48,49,53, and 55 rejected under 35 U.S.C. 103(a) as being unpatentable over Hanrahan in view of US Patent No. 6,543,002 of Kahle et al. referred hereinafter "Kahle".

In regards to claim 29, Hanrahan discloses a program storage device readable by a computer, the program storage device tangibly embodying one or more programs of instructions executable by the computer to perform operations for determining when to perform an error recovery instruction, the operations comprising:

receiving an error recovery instruction. Hanrahan discloses a receiving a quiesce signal (see column 9 line 13-14).

monitoring a processor interface to identify processor status for determining a time to perform the error recovery instruction for withholding access to the local processor. Hanrahan

discloses determining when there are no longer bus requests (see column 7 lines 20-23), indicating monitoring a processor interface, and performing no further activity (see column 7 lines 20-23), indicating withholding access to the local processor.

performing the error recovery instruction when the monitoring determines a time for performing the error recovery instruction. Hanrahan discloses sequencing to a known state for retry, recover, and debug operations (see column 2 lines 24-26).

However, Hanrahan fails to disclose:

beginning a timeout task;

Kahle disclose a hang detection unit for recovering from a hang condition via a hang recovery sequence. Kahle further discloses a hang condition is determined when the maximum number of clock cycles since the most recent assertion of completion valid signal is exceeded (see column 6 lines 40-45), indicating a timeout task.

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of Hanrahan and Kahle to have a hang detection unit for recovering from a hang condition via a hang recovery sequence, wherein a hang condition is determined when the maximum number of clock cycles since the most recent assertion of completion valid signal is exceeded, thus indicating a timeout task. A person of ordinary skill in the art would have been motivated to combine the teachings because Hanrahan is concerned with recovering from system errors (see column 1 lines 23-25), and incorporating a hang detection unit, as per teachings of Kahle, enables recovery from errors resulting from hang conditions (see column 6 lines 40-45).

In regards to claim 30, Hanrahan in view of Kahle discloses the claim limitations as discussed above. Kahle further discloses forcing an execution of the error recovery instruction when the timeout task expires before the monitoring determines a time to perform the error recovery instruction. Kahle discloses performing a hang recovery sequence to recover from hang conditions when the maximum number of clock cycles since the most recent assertion of completion valid signal is exceeded (see column 6 lines 40-45).

In regards to claim 31, Hanrahan in view of Kahle discloses the claim limitations as discussed above. Hanrahan further discloses resuming normal operations after performing the error recovery instruction. Hanrahan discloses resuming operations following the quiesce state (see column 9 lines 24-26).

In regards to claim 32, Hanrahan in view of Kahle discloses the claim limitations as discussed above. Hanrahan further discloses

monitoring a processor interface to a host bus for an idle condition. Hanrahan discloses determining when there are no longer bus requests (see column 7 lines 20-23).

withholding access to the processor interface when the idle condition is detected. Hanrahan discloses performing no further activity (see column 7 lines 20-23).

after access to the processor interface is withheld, interrogating all data transfer paths to determine when all the data paths are idle. Hanrahan discloses waiting for all operations to complete before sequencing to a known state (see column 9 lines 15-20).

identifying the time to perform the error recovery instruction when all data transfer paths are idle. Hanrahan discloses sequencing to a known state when all current operations that have completed (see column 9 lines 15-22).

In regards to claim 33, Hanrahan in view of Kahle discloses the claim limitations as discussed above. Hanrahan further discloses resuming normal operations after performing the error recovery instruction. Hanrahan discloses resuming operations following the quiesce state (see column 9 lines 24-26).

In regards to claim 34, Hanrahan in view of Kahle discloses the claim limitations as discussed above. Hanrahan further discloses resuming normal operations after performing the error recovery instruction. Hanrahan discloses resuming operations following the quiesce state (see column 9 lines 24-26).

In regards to claim 36, Hanrahan discloses the claim limitations as discussed above.

However, Hanrahan fails to disclose beginning a timeout task after receiving the error recovery instruction and forcing an execution of the error recovery instruction when the timeout task expires before the monitoring determines a time to perform the error recovery instruction.

Kahle discloses determining when the maximum number of clock cycles since the most recent assertion of completion valid signal is exceeded for a hang condition (see column 6 lines 40-45), indicating a timeout task. Kahle further disclose a hang detection unit for performing a hang recovery sequence to recover from hang conditions when the maximum number of clock cycles since the most recent assertion of completion valid signal is exceeded (see column 6 lines 40-45), indicating forcing an execution of the error recovery instruction when the timeout task expires before the monitoring determines a time to perform the error recovery instruction.

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of Hanrahan and Kahle to have a hang detection unit for

recovering from a hang condition via a hang recovery sequence, wherein a hang condition is determined when the maximum number of clock cycles since the most recent assertion of completion valid signal is exceeded, thus indicating a timeout task after receiving the error recovery instruction and forcing an execution of the error recovery instruction when the timeout task expires before the monitoring determines a time to perform the error recovery instruction. A person of ordinary skill in the art would have been motivated to combine the teachings because Hanrahan is concerned with recovering from system errors (see column 1 lines 23-25), and incorporating a hang detection unit, as per teachings of Kahle, enables recovery from errors resulting from hang conditions (see column 6 lines 40-45).

In regards to claim 37, Hanrahan in view of Kahle discloses the claim limitations as discussed above. Hanrahan further discloses resuming normal operations after performing the error recovery instruction. Hanrahan discloses resuming operations following the quiesce state (see column 9 lines 24-26).

In regards to claim 41, Hanrahan discloses an apparatus for quiescing processor control logic upon receipt of an error recovery instruction, comprising:

self-quiesce logic for receiving an error recovery instruction. Hanrahan discloses a receiving a quiesce signal (see column 9 line 13-14).

begins to monitor a processor interface to identify processor status for determining a time to perform the error recovery instruction for withholding access to the local processor. Hanrahan discloses determining when there are no longer bus requests (see column 7 lines 20-23),

indicating monitoring a processor interface, and performing no further activity (see column 7 lines 20-23), indicating withholding access to the local processor.

performs the error recovery instruction when the monitoring determines a time for performing the error recovery instruction. Hanrahan discloses sequencing to a known state for retry, recover, and debug operations (see column 2 lines 24-26).

However, Hanrahan fails to explicitly disclose:

a timer, coupled ot the self-quiecse logic, for determining when to force execution of the error recovery instruction, wherein the self-quiesce logic initiates the timer when the error recovery instruction is received.

Kahle disclose a hang detection unit for recovering from a hang condition via a hang recovery sequence, wherein a hang condition is determined when the maximum number of clock cycles since the most recent assertion of completion valid signal is exceeded (see column 6 lines 40-45), indicating a timer for determining when to force execution of the error recovery instruction, wherein the self-quiesce logic initiates the timer when the error recovery instruction is received.

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of Hanrahan and Kahle to have a hang detection unit for recovering from a hang condition via a hang recovery sequence, wherein a hang condition is determined when the maximum number of clock cycles since the most recent assertion of completion valid signal is exceeded, thus indicating a timer for determining when to force execution of the error recovery instruction, wherein the self-quiesce logic initiates the timer when the error recovery instruction is received. A person of ordinary skill in the art would have

been motivated to combine the teachings because Hanrahan is concerned with recovering from system errors (see column 1 lines 23-25), and incorporating a hang detection unit, as per teachings of Kahle, enables recovery from errors resulting from hang conditions (see column 6 lines 40-45).

In regards to claim 42, Hanrahan in view of Kahle discloses the claim limitations as discussed above. Kahle further discloses wherein the self-quiesce logic forces an execution of the error recovery instruction when the timer expires before the self-quiesce logic determines a time to perform the error recovery instruction. Kahle discloses performing a hang recovery sequence to recover from hang conditions when the maximum number of clock cycles since the most recent assertion of completion valid signal is exceeded (see column 6 lines 40-45).

In regards to claim 43, Hanrahan in view of Kahle discloses the claim limitations as discussed above. Hanrahan further discloses wherein the self-quiesce logic allows resuming normal operations after the error recovery instruction is performed. Hanrahan discloses resuming operations following the quiesce state (see column 9 lines 24-26).

In regards to claim 44, Hanrahan in view of Kahle discloses the claim limitations as discussed above. Hanrahan further discloses

wherein the self-quiesce logic monitors a processor interface to a host bus to identify processor status for determining a time to perform the error recovery instruction withholding access to the local processor by monitoring a processor interface for an idle condition. Hanrahan discloses determining when there are no longer bus requests (see column 7 lines 20-23).

withholding access to the processor interface when the idle condition is detected. Hanrahan discloses performing no further activity (see column 7 lines 20-23).

after access to the processor interface is withheld, interrogating all data transfer paths to determine when all the data paths are idle. Hanrahan discloses waiting for all operations to complete before sequencing to a known state (see column 9 lines 15-20).

identifying the time to perform the error recovery instruction when all data transfer paths are idle. Hanrahan discloses sequencing to a known state when all current operations that have completed (see column 9 lines 15-22).

In regards to claim 45, Hanrahan in view of Kahle discloses the claim limitations as discussed above. Hanrahan further discloses wherein the self-quiesce logic allows resuming normal operations after the error recovery instruction is performed. Hanrahan discloses resuming operations following the quiesce state (see column 9 lines 24-26).

In regards to claim 46, Hanrahan in view of Kahle discloses the claim limitations as discussed above. Hanrahan further discloses wherein the self-quiesce logic allows resuming normal operations after the error recovery instruction is performed. Hanrahan discloses resuming operations following the quiesce state (see column 9 lines 24-26).

In regards to claim 48, Hanrahan discloses the claim limitations as discussed above. However, Hanrahan fails to explicitly disclose:

a timer for determining when to abort the monitoring of processor status and data path activity and cause an execution of the error recovery instruction.

Kahle disclose a hang detection unit for recovering from a hang condition via a hang recovery sequence, wherein a hang condition is determined when the maximum number of clock cycles since the most recent assertion of completion valid signal is exceeded (see column 6 lines

40-45), indicating a timer for determining when to abort the monitoring of processor status and data path activity and cause an execution of the error recovery instruction.

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of Hanrahan and Kahle to have a hang detection unit for recovering from a hang condition via a hang recovery sequence, wherein a hang condition is determined when the maximum number of clock cycles since the most recent assertion of completion valid signal is exceeded (see column 6 lines 40-45), thus indicating a timer for determining when to abort the monitoring of processor status and data path activity and cause an execution of the error recovery instruction. A person of ordinary skill in the art would have been motivated to combine the teachings because Hanrahan is concerned with recovering from system errors (see column 1 lines 23-25), and incorporating a hang detection unit, as per teachings of Kahle, enables recovery from errors resulting from hang conditions (see column 6 lines 40-45).

In regards to claim 49, Hanrahan in view of Kahle discloses the claim limitations as discussed above. Hanrahan further discloses wherein the self-quiesce logic causes normal operations to be resumed after performing the error recovery instruction. Hanrahan discloses resuming operations following the quiesce state (see column 9 lines 24-26).

In regards to claim 53, Hanrahan in view of Kahle discloses a method for determining when to perform an error recovery instruction, comprising:

receiving an error recovery instruction. Hanrahan discloses a receiving a quiesce signal (see column 9 line 13-14).

monitoring processor interface to identify processor status for determining a time to perform the error recovery instruction for withholding access to the local processor. Hanrahan discloses determining when there are no longer bus requests (see column 7 lines 20-23), indicating monitoring a processor interface, and performing no further activity (see column 7 lines 20-23), indicating withholding access to the local processor.

performing the error recovery instruction when the monitoring determines a time for performing the error recovery instruction. Hanrahan discloses sequencing to a known state for retry, recover, and debug operations (see column 2 lines 24-26).

However, Hanrahan fails to disclose:

beginning a timeout task;

Kahle disclose a hang detection unit for recovering from a hang condition via a hang recovery sequence. Kahle further discloses a hang condition is determined when the maximum number of clock cycles since the most recent assertion of completion valid signal is exceeded (see column 6 lines 40-45), indicating a timeout task.

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of Hanrahan and Kahle to have a hang detection unit for recovering from a hang condition via a hang recovery sequence, wherein a hang condition is determined when the maximum number of clock cycles since the most recent assertion of completion valid signal is exceeded, thus indicating a timeout task. A person of ordinary skill in the art would have been motivated to combine the teachings because Hanrahan is concerned with recovering from system errors (see column 1 lines 23-25), and incorporating a hang detection

unit, as per teachings of Kahle, enables recovery from errors resulting from hang conditions (see column 6 lines 40-45).

In regards to claim 55, Hanrahan in view of Kahle discloses an apparatus for quiescing processor control logic upon receipt of an error recovery instruction, comprising:

means for receiving an error recovery instruction. Hanrahan discloses a receiving a quiesce signal (see column 9 line 13-14).

wherein the means for receiving the error recovery instruction begins to monitor a processor interface to identify processor status for determining a time to perform the error recovery instruction for withholding access to the local processor. Hanrahan discloses determining when there are no longer bus requests (see column 7 lines 20-23), indicating monitoring a processor interface, and performing no further activity (see column 7 lines 20-23), indicating withholding access to the local processor.

performs the error recovery instruction when a time for performing the error recovery instruction is determined. Hanrahan discloses sequencing to a known state for retry, recover, and debug operations (see column 2 lines 24-26).

However, Hanrahan fails to disclose:

initiating a timer when the error recovery instruction is received and means for determining when to force execution of the error recovery instruction.

Kahle discloses determining when the maximum number of clock cycles since the most recent assertion of completion valid signal is exceeded for a hang condition (see column 6 lines 40-45), indicating initiating a timer when the error recovery instruction is received. Kahle

further disclose a hang detection unit for performing a hang recovery sequence to recover from hang conditions when the maximum number of clock cycles since the most recent assertion of completion valid signal is exceeded (see column 6 lines 40-45), indicating means for determining when to force execution of the error recovery instruction.

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of Hanrahan and Kahle to have a hang detection unit for recovering from a hang condition via a hang recovery sequence, wherein a hang condition is determined when the maximum number of clock cycles since the most recent assertion of completion valid signal is exceeded, thus indicating initiating the timer when the error recovery instruction is received and means for determining when to force execution of the error recovery instruction. A person of ordinary skill in the art would have been motivated to combine the teachings because Hanrahan is concerned with recovering from system errors (see column 1 lines 23-25), and incorporating a hang detection unit, as per teachings of Kahle, enables recovery from errors resulting from hang conditions (see column 6 lines 40-45).

Response to Arguments

Applicant's arguments filed April 5, 2007 have been fully considered but they are not persuasive.

In response to applicant's argument on page 13 of Remarks, "Accordingly, Mito et al. does not monitor a processor interface to identify processor status for determining a time to perform the error recovery instruction. Mito et al. also fails to determine a time to perform the error recovery instruction for withholding access to the local processor. Further, Mito et al. also

fails to perform the error recovery instruction for withholding access to the local processor when the monitoring determines a time for performing the error recovery instruction," examiner respectfully disagrees.

As stated in the rejection above, Mito discloses monitoring for interrupts from a power management processor interface to initiate a suspend routine (see figure 2,4 and column 8 lines 25-30 and 45-47), indicating monitoring a processor interface to identify processor status for determining a time to perform the error recovery instruction. Mito further discloses initiating a suspend routine (see column 8 lines 45-47), indicating determining a time for performing the error recovery instruction for withholding access to the local processor when the monitoring determines a time for performing the error recovery instruction. Argument is moot. Examiner maintains his rejection.

In response to applicant's argument on page 14-16 of Remarks, "Accordingly, Hanrahan et al. and Kahle et al. does not monitor a processor interface to identify processor status for determining a time to perform the error recovery instruction. Hanrahan et al. and Kahle et al. also fails to determine a time to perform the error recovery instruction for withholding access to the local processor. Further, Hanrahan et al. and Kahle et al. also fails to perform the error recovery instruction for withholding access to the local processor when the monitoring determines a time for performing the error recovery instruction," examiner respectfully disagrees.

As stated in the rejection above, Hanrahan discloses determining when there are no longer bus requests to the bus arbiter logic or processor interface (see column 7 lines 20-23), indicating monitor a processor interface to identify processor status for determining a time to

perform the error recovery instruction. Hanrahan also discloses performing no further activity (see column 7 lines 20-23), indicating determining a time to perform the error recovery instruction for withholding access to the local processor. Hanrahan also discloses sequencing to a known state for retry, recover, and debug operations (see column 2 lines 24-26), indicating perform the error recovery instruction for withholding access to the local processor when the monitoring determines a time for performing the error recovery instruction. Argument is moot. Examiner maintains his rejection.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Emerson C. Puente whose telephone number is (571) 272-3652. The examiner can normally be reached on 8-5 M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Robert W. Beausoliel can be reached on (571) 272-3645. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Emerson Puente

Emerson Puente
Examiner
AU 2113